UNIVERSITY EXAMINATION TIMETABLING: A GENERAL MODEL

Muhammad Rozi Malim[1], (Ahamad Tajudin Khader[2], Adli Mustafa[3])

[1]School of Computer Science, University of Science Malaysia, 11800 Minden, Penang, Malaysia
malimmr@celcom.net.my

[2]School of Computer Science, University of Science Malaysia, 11800 Minden, Penang, Malaysia
tajudin@cs.usm.my

[3]School of Mathematical Science, University of Science Malaysia, 11800 Minden, Penang, Malaysia
adli@cs.usm.my

Abstract: The university examination timetabling is known to be a highly constrained combinatorial optimization problem. Various examination timetabling models have been described in the literature but most of the models were developed for isolated problems. This paper discusses the university examination timetabling in general. The works on examination timetabling available in the literature are studied, and all constraints are gathered. The ultimate goal is to develop a general model of university examination timetabling problem. The general model is presented; each of the constraints (hard and soft) is mathematically formulated as a 0-1 integer programming. For future work, this model will be used to develop a general model for a more complex timetabling problem, university course timetabling.

Keywords: Examination timetabling; timetabling constraints; general model.

1. Introduction

Timetabling problems arise in a wide variety of domains; education, transport, healthcare institutions, etc. [Burke et al. (2001)]. The university timetabling problems have attracted the attention of scientists from a number of differing disciplines, including operations research and artificial intelligence, since the 1950s. *University examination timetabling problem* (UETP) is a specific case of the more general timetabling problem and known to be highly constrained combinatorial optimization problem.

The construction of an examination timetable is a common problem for all institutions of higher education. Given a set of exams, a set of students, a set of timeslots, a set of rooms, and a set of student enrollments, the problem is to assign exams to timeslots and rooms subject to a set of *hard* and *soft* constraints. The main difficulty is to obtain a *conflict-free* schedule within a limited number of timeslots and rooms. Conflicting objectives and the changing set of constraints in different institutions makes the examination timetabling problem (ETP) very challenging. Various university examination timetabling models have been described in the literature, but most of the models were developed for isolated problems, i.e. problem-based models.

This paper discusses the UETP in general. The works on examination timetabling available in the literature are studied, and all constraints (hard and soft) are gathered. The ultimate goal is to develop a general model of the UETP. This general model is presented in Section 4; all constraints (hard and soft) are mathematically formulated as *0-1 integer programming.*

2. Examination Timetabling Constraints

The sets of constraints differ significantly from institution to institution. Different categories of people have different priorities in the timetabling process and are affected differently by its outcome [Burke et al. (2001)]. In the examination timetabling process, there are three main groups of people who are affected by the results of the process; administration, departments, and students. Consequently, the quality of a timetable can be assessed from various points of view and the importance of a particular constraint can very much depend upon the priorities of the three categories.

In Burke et al. [1995], from the survey conducted on 95 British Universities, the following constraints are observed, sorted in ascending order according to their importance:

(1) There must not be more students scheduled to a room than there are seats.
(2) Exams with questions in common must be scheduled in the same timeslot.
(3) Some exams may only be scheduled within a particular set of timeslots.
(4) Only exams of the same length may be scheduled in the same room.
(5) Exams with the most students must be scheduled early in the timetable.
(6) Some exams must only take place in particular rooms.

(7) Large exam halls must be scheduled in preference to smaller ones.
(8) Exam A must be scheduled before exam B.
(9) No student is scheduled to exams in two consecutive timeslots.
(10) No student is scheduled to more than one exam in any particular day.
(11) Each student's exams must be evenly spread throughout the timetable.
(12) No student is scheduled to exams in two consecutive days.
(13) Exams must be scheduled to rooms near to the relevant department.

The basic hard constraint in an ETP is 'no student should have two exams in the same timeslot', i.e. no student conflicts. Burke et al. [2001] and Cote et al. [2004] considered this as the only hard constraint in their ETPs. However, some institutions allow a student to have two exams in the same timeslot, as long as an appropriate arrangement can be made. Erben and Song [2004], and Gambhava and Sanghani [2003] considered two hard constraints: (1) no student can be in more than one place at a time, and (2) there must be sufficient seats to house all the students. In Burke et al. [2003], the three main hard constraints are: (1) every exam must be assigned to exactly one timeslot, (2) no individual should be in two different places at once, and (3) there must be sufficient resources available in each timeslot for all exams timetabled.

The common soft constraints for ETPs that agreed by most researchers in the literature are as follows:
(1) A student should not be expected to sit two exams in adjacent timeslots.
(2) No students should have to take two exams on the same day.
(3) The total number of students in the same timeslot must be less than the capacity of the timeslot.
(4) The total number of students in the same room must be less than the capacity of the room.
(5) The total number of exams in the same timeslot must be less than some specified number.
(6) Larger exams come first to allow more time for them to be marked.
(7) Some exams may need to be scheduled before, after or at the same time as another.
(8) Some exams need to be preassigned to a timeslot or can only be assigned to a subset of timeslots.
(9) Some exams may require specific rooms.
(10) The length of timetable.
(11) Only exams of similar length are scheduled at the same timeslot in the same room.
(12) Certain groups of exams may be required to take place at the same time (common questions).
(13) Certain exams are required to be on the same day.
(14) Certain timeslots are excluded for an examination.
(15) Some rooms are only available in specific timeslots.

3. Examination Timetabling Models

Some of the examination timetabling models, described in the literature, are presented as follows:

In Paquete and Fortseca [2001], an exam timetable of a typical educational institution can be represented as a mapping $t: E \rightarrow T$, where E is a set of exams and T is a set of timeslots. An ETP (E, T, f) is the problem of finding an optimal mapping $t_{opt} \in H$, where H denotes the set of all possible mappings from E to T, such that, given a cost function $f: H \rightarrow R$, $f(t_{opt}) \leq f(t)$, $\forall\ t \in H$. Most evolutionary algorithm approaches to timetabling formulate f as a weighted sum of numbers of constraint violations, where the weights are chosen according to the importance of the corresponding constraints.

Merlot et al. [2002] defined the constraint programming model, to find a first feasible timetable, as: $E = \{1,\ldots,n\}$ denotes the given set of n exams; s_i is the number of students in exam i; $T = \{1,\ldots,v\}$ denotes the set of v timeslots; $R \subset E \times T$ represents a set of given exam-timeslot restrictions, so that $(a, b) \in R$ indicates that timeslot b cannot be allocated to exam a; C_t is the total capacity (of students) of timeslot t; D_{ij} is the number of students enrolled in both exams i and j; and variable $x_i \in T$ indicates the timeslot allocated to exam i. The timeslots are assumed to be time ordered. They also defined the domain of each variable x_i to be the set of all timeslots that can feasibly be allocated to exam i.

Wong et al. [2002] characterized the final ETP as follows: There are q courses $c_1,\ldots,c_q$ with one exam for each course. These exams are partitioned into r exam-groups $G_1,\ldots,G_r$ such that for each G_i there are students that take all exams belonging to group G_i. Every school day of the semester has 3 class-timeslots. Each course c_i is given a value such that $c_i \in \{0,1,2\}$ that identifies the membership of c_i to one of the 3 class-timeslots. For the exam schedule, there are d days of exams. Thus, a total of $3d$ exam-timeslots are available for exam assignment. To take into account the availability of exam-timeslots and the possibility of preassignment, they defined an availability matrix $\mathbf{A}_{q\times 3d}$ such that $a_{i,k}$=1 if timeslot k is available for exam i, and 0 otherwise. Using a similar logic, they defined a preassignment matrix $\mathbf{B}_{q\times 3d}$ such that $b_{i,k}$=1 if exam i is preassigned to timeslot k, 0 otherwise. Finally, they denoted l_k as the maximum number of exam for timeslot k. The natural candidate solution representation is a $1\times q$ vector. Each element of the vector represents an exam. The vector element's value represents the exam-timeslot assigned to that exam. For population of candidate solutions, a population matrix $\mathbf{X}=[x_{i,j}]$ is defined (i=1,...,N, j=1,...,q), where N is the number of candidate solutions. In another paper, Wong et al. [2004], the following timetabling model was used: Given a set of exams $E=\{e_1,\ldots,e_{|E|}\}$ and a set of timeslots $T=\{1, 2,\ldots,|T|\}$, the goal of the timetabling procedure is to obtain an assignment $T\Rightarrow E$ where each exam in E is allocated to a timeslot in T. The result of such an assignment is a timetable represented by a set h of ordered couples (t, e) where $t\in T$ and $e\in E$. A timetable h is called feasible if it satisfies all required constraints.

Kendall and Mohd Hussin [2004] represented the ETP at UiTM as: E is a set of m exams $E_1,E_2,\ldots,E_m$; S is a set of n timeslots $S_1,S_2,\ldots,S_n$; U is a set of u campuses $U_1,U_2,\ldots,U_u$. A final exam timetable T_{mn} such that T_{ik} = 1 if exam i scheduled in timeslot k, 0 otherwise; $CampusType = \{A,B\}$ where campus type A has half-day Saturday and full-day Sunday weekend, and campus type B has half-day Thursday and full-day Friday weekend. A conflict matrix $\mathbf{C}_{mm}$ such that C_{ij} = total number of students sitting for both exams i and j categorized by campus type. A co-schedule matrix $\mathbf{R}_{mm}$ such that R_{ik} = 1 if exam i and exam k must be scheduled in the same timeslot, 0 otherwise.

4. General Model of University Examination Timetabling

Here, a *general university examination timetabling model* (GUETM) is presented. This model is based on the models, and constraints, discussed in the literature. Each of the constraints (hard and soft) is mathematically formulated as a *0-1 integer programming*. UETP is known to be a highly constrained and complex optimization problem. To reduce the complexity of the problem, it is normally divided into *two subproblems*; 'exam-timeslot assignment' and 'exam-room assignment'. These two subproblems can be modeled separately. In 'exam-timeslot assignment' exams are scheduled into a fixed number of timeslots, and in 'exam-room assignment' exams (in each timeslot) are assigned to rooms. Hence, in a UETP, an *assignment* is an ordered *3-tuple* (a, b, c), where $a\in E$, $b\in T$, $c\in R$, and has the straightforward general interpretation: "exam a starts at timeslot b in room c". The main components of a university examination timetabling model are *problem definition and initialization*, *hard constraints*, and *soft constraints*.

4.1. *Problem definition and initialization*

There are *six* sets of variables that should be taken into account in a university examination timetabling:

Exam: represents the exam to be timetabled. The domain of this variable, E, is the set of all exams to be scheduled. Each exam e_i, $i\in\{1,\ldots,n_1\}$, has student enrollments, department, length, and type.

Timeslot: represents the time occupied by the exam. The domain of this variable, T, is the set of all timeslots in the timeslot system. Each timeslot t_j, $j\in\{1,\ldots,n_2\}$, has start & finish times, and type.

Room: represents the room where the exam to be held. The domain of this variable, R, is the set of all rooms available for the exams. Each room r_i, $i\in\{1,\ldots,n_3\}$, has department, size, and type.

Student: represents the student enrolled for the exams. The domain of this variable, S, is the set of all students enrolled for the exams. Each student s_j, $j\in\{1,\ldots,n_4\}$, has department, & a list of exams.

Department: represents the department or faculty. The domain of this variable, D, is the set of all departments or faculties at an institution. Each department d_i, $i\in\{1,\ldots,n_5\}$, has a list of exams, a list of rooms, a list of students, and a list of invigilators.

Invigilator: represents the lecturer, tutor or administrator who invigilates the exam. The domain of this variable, I, is the set of all invigilators for the examination. Each invigilator i_j, $j\in\{1,\ldots,n_6\}$, has department, a list of restricted exams, and a list of timeslots during which he/she is unavailable.

The matrices are required to show the interrelationships between these sets of variables. These matrices would assist the formulation of the hard and soft constraints as 0-1 integer programming. There are two types of matrices, *input matrices* and *output matrices*. The input matrices are the matrices where the values are known earlier (timetabling data), and have been allocated or preassigned. The output matrices are the *assignment* matrices where the values need to be determined by solving the ETP.

4.1.1. *Input matrices*

Exam-timeslot preassignment matrix: some exams must be scheduled in some specific timeslots; **A**=$E\times T$, a_{ij}=1 if exam e_i is preassigned to timeslot t_j, 0 otherwise; $n_p(e_i)\leq 1$ for all i since each exam may be preassigned to only one timeslot; $n_q(t_j)$ represents the number of preassigned exams in timeslot t_j.
Exam-timeslot restriction matrix: some exams must *not* be scheduled in some specific timeslots; **B**=$E\times T$, b_{ij}=1 if exam e_i is restricted to timeslot t_j, 0 otherwise; $n_t(e_i)$ is the number of restricted timeslots for exam e_i; $n_e(t_j)$ is the number of restricted exams for timeslot t_j.
Exam-room preassignment matrix: some exams must be held in some specific rooms; **D**=$E\times R$, d_{ij}=1 if exam e_i is preassigned to room r_j, 0 otherwise; $n_u(e_i)$ is the number of preassigned rooms for exam e_i; $n_v(r_j)$ represents the number of preassigned exams in room r_j.
Exam-student enrollment matrix: **E**=$E\times S$, e_{ij}=1 if exam e_i is enrolled by student s_j, 0 otherwise; $n_s(e_i)$ represents the number of students enrolled for exam e_i; $n_e(s_j)$ is the number of exams enrolled by student s_j; $\Sigma n_s(e_i)=\Sigma n_e(s_j)=N$ for all i and j, and N is the total number of student enrollments.
Exam-department matrix: **H**=$E\times D$, h_{ij}=1 if exam e_i belongs to department d_j, 0 otherwise; $n_d(e_i)$ is the number of departments share the same exam e_i; $n_e(d_j)$ is the number of exams belong to department d_j.
Room-timeslot availability matrix: **J**=$R\times T$, j_{ij}=1 if room r_i is available in timeslot t_j, 0 otherwise; $n_t(r_i)$ is the number of timeslots in which room r_i is available; $n_r(t_j)$ is the number of rooms available in timeslot t_j.
Room-department matrix: **K**=$R\times D$, k_{ij}=1 if room r_i is located at (or allocated to) department d_j, 0 otherwise; $n_d(r_i)$ is the number of departments share the same room r_i; $n_r(d_j)$ is the number of rooms located at (or allocated to) department d_j.
Invigilator-exam restriction matrix: an invigilator who was the lecturer or tutor for some courses must not invigilate the exams for those courses; **L**=$I\times E$, l_{ij}=1 if invigilator i_i is restricted to exam e_j, 0 otherwise; $n_e(i_i)$ is the number of restricted exams for i_i; $n_i(e_j)$ is the number of invigilators that restricted to exam e_j.
Invigilator-timeslot availability matrix: for some reasons, an invigilator is unavailable during certain timeslots; **M**=$I\times T$, m_{ij}=1 if invigilator i_i is available during timeslot t_j, 0 otherwise; $n_t(i_i)$ is the number of timeslots during which i_i is available; $n_i(t_j)$ is the number of invigilators that available during t_j.
Student-conflict matrix: **C**=$E\times E$, c_{ij} is the number of students taking both exams e_i and e_j. This matrix is required to avoid or minimize the number of students having two exams in the same timeslot.

We also have the *timeslot-capacity* of students for each timeslot (based on *room-timeslot availability*), $n_c(t_i)$; the *room-capacity* of students for each room, $n_c(r_j)$; the *exam-capacity* for each timeslot (based on *room-timeslot availability*), $n_{ce}(t_i)$; the *exam-capacity* for each room (based on *room-capacity*), $n_{ce}(r_j)$; the *invigilator-capacity* for each timeslot (based on *room-timeslot availability*), $n_{ci}(t_i)$; the *invigilator-capacity* for each room (based on *room-capacity*), $n_{ci}(r_j)$; the minimum and maximum number of times of invigilation for each invigilator, $n_{\min}(i_k)$ and $n_{\max}(i_k)$; and the length of each exam (in hours), $l(e_i)$.

4.1.2. *Output matrices*

The following *(four)* output matrices will form a complete university examination timetable.
Exam-timeslot assignment matrix: **Q**=$E\times T$, q_{ij}=1 if exam e_i is assigned to timeslot t_j, 0 otherwise; $n_t(e_i)$=1 for all i, i.e. each exam must be assigned to only one timeslot; $n_e(t_j)$ is the number of exams assigned to t_j.
Exam-room assignment matrix: **R**=$E\times R$, r_{ij}=1 if exam e_i is assigned to room r_j, 0 otherwise; $n_r(e_i)$ is the number of rooms assigned to exam e_i; $n_e(r_j)$ is the number of exams assigned to room r_j.
Invigilator-timeslot assignment matrix: **S**=$I\times T$, s_{ij}=1 if invigilator i_i is assigned to timeslot t_j, 0 otherwise; $n_t(i_i)$ is the number of timeslots assigned to invigilator i_i; $n_i(t_j)$ is the number of invigilators assigned to t_j.
Invigilator-room assignment matrix: **T**=$I\times R$, t_{ij}=1 if invigilator i_i is assigned to room r_j, 0 otherwise; $n_i(r_j)$ is the number of invigilators assigned to r_j; $n_r(i_i)$ is the number of rooms assigned to invigilator i_i.

4.2. *Hard constraints and mathematical formulation*

Hard constraints must be satisfied in order to produce a *feasible* timetable. Any timetable which fails to satisfy all these constraints is deemed to be *infeasible*. The hard constraints for the two subproblems (*exam-timeslot assignment* and *exam-room assignment*) will be considered separately. Each institution will apply some or all of these hard constraints. However, each institution will have some unique combination of hard constraints, as policies differ from institution to institution.

4.2.1. *Hard constraints for exam-timeslot assignment*

The (*eight*) common hard constraints for 'exam-timeslot assignment' (ordered by their importance):

H1) *Exam-clashing*: Every exam must be assigned to exactly one timeslot of the timetable; formulated using the matrix **Q**, where $x(e_i, t_j) = 0$ if $\sum_{j=1}^{n_2} q_{ij} = 1$, and 1 otherwise;

$$\sum_{i=1}^{n_1} x(e_i, t_j) = 0. \tag{1}$$

H2) *Student-clashing*: No student should be scheduled in two different places at once, i.e. any two exams which have students in common must not be scheduled both in the same timeslot; formulated using the matrices **C** and **Q**, where c_{ij} is the number of students taking both exams e_i and e_j;

$$\sum_{k=1}^{n_2} \sum_{i=1}^{n_1-1} \sum_{j=i+1}^{n_1} c_{ij} \cdot q_{ik} \cdot q_{jk} = 0. \tag{2}$$

H3) *Common-question*: Certain exams with questions in common must be grouped and scheduled together in the same timeslot. For instance, if exams e_i and e_j have common questions, then both must be assigned in the same timeslot; formulated using the matrix **Q**, where $x(e_i, e_j) = 0$ if $q_{ik}=q_{jk}$, and 1 otherwise, for each group of two exams with common questions,

$$\sum_{k=1}^{n_2} x(e_i, e_j) = 0. \tag{3}$$

H4) *Timeslot-capacity*: The total number of students in all exams in the same timeslot must be less than the total capacity for that timeslot; formulated using the matrices **E** and **Q**, where $x_c(e_i, t_j) = 0$ if $\sum_{i=1}^{n_1} n_s(e_i) \cdot q_{ij} \leq n_c(t_j)$, and 1 otherwise,

$$\sum_{j=1}^{n_2} x_c(e_i, t_j) = 0. \tag{4}$$

H5) *Preassigned-timeslot*: Some exams must be assigned to specific timeslots as in exam-timeslot preassignment matrix; formulated using the matrices **A** and **Q**,

$$\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} a_{ij} \cdot (1 - q_{ij}) = 0. \tag{5}$$

H6) *Restricted-timeslot*: Some exams are restricted to certain timeslots as in exam-timeslot restriction matrix; formulated using the matrices **B** and **Q**,

$$\sum_{j=1}^{n_2} \sum_{i=1}^{n_1} b_{ij} \cdot q_{ij} = 0. \tag{6}$$

H7) *Invigilator-exam restriction*: An invigilator who was the lecturer or tutor for a course with exam e_i must not invigilate during the timeslot assigned to the exam; formulated using the matrices **L, Q & S,**

$$\sum_{i=1}^{n_6} \sum_{j=1}^{n_1} \sum_{k=1}^{n_2} l_{ij} \cdot q_{jk} \cdot s_{ik} = 0. \tag{7}$$

H8) *Invigilator-availability*: Some invigilators, for some reasons, are unavailable during certain timeslots; formulated using the matrices **M** and **S**,

$$\sum_{j=1}^{n_2} \sum_{i=1}^{n_6} (1 - m_{ij}) \cdot s_{ij} = 0. \tag{8}$$

Most researchers in the literature considered only the first *two* hard constraints. Some institutions consider the constraint (H2) as a soft constraint (S1) as long as an appropriate arrangement can be made. The constraints (H4), (H5) and (H6) may also be considered as soft constraints (S2), (S3) and (S4). The constraint (H7) may be considered as a soft constraint (S5) but an appropriate arrangement must be made.

4.2.2. *Hard constraints for exam-room assignment*

The (*four*) common hard constraints for 'exam-room assignment' (ordered by their importance):

H9) *Room-capacity*: There must be sufficient seats available in each room for all exams timetabled; formulated using **E**, **Q** and **R**, where $x_c(e_i,t_j,r_k)=0$ if $\sum_{i=1}^{n_1} n_s(e_i)\cdot q_{ij}\cdot r_{ik} \le n_c(r_k)$, and 1 otherwise,

$$\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_c(e_i,t_j,r_k)=0\,. \qquad (9)$$

H10) *Preassigned-room*: Some exams must be assigned to special rooms as in exam-room preassignment matrix; formulated using the matrices **D** and **R**,

$$\sum_{i=1}^{n_1}\sum_{j=1}^{n_3} d_{ij}\cdot(1-r_{ij})=0\,. \qquad (10)$$

H11) *Exam-room availability*: An exam can only be assigned to room(s) if and only if the room is available; formulated using the matrices **J** and **R**,

$$\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} r_{ik}\cdot(1-j_{kj})=0\,. \qquad (11)$$

H12) *Invigilator-clashing*: No invigilator should be scheduled in two different rooms at once; formulated using the matrices S and T, where $x(i_i,t_j,r_k)=0$ if $\sum_{k=1}^{n_3} s_{ij}\cdot t_{ik}\le 1$, and 1 otherwise,

$$\sum_{i=1}^{n_6}\sum_{j=1}^{n_2} x(i_i,t_j,r_k)=0\,. \qquad (12)$$

The hard constraints (H9), (H10) and (H11), for some reasons, may also be considered as soft constraints (S16), (S17) and (S18), respectively.

4.3. *Soft constraints and mathematical formulation*

Soft constraints are those which are desirable to be satisfied, but which in general cannot all be wholly met. Soft constraints are generally more numerous and varied, and far more dependent on the needs of the individual problem than the more obvious hard constraints. The *quality* of a resulting timetable can be measured according to an objective function which weights the violation of the soft constraints. The soft constraints for the two subproblems will be considered separately. Each institution will apply some or all of these soft constraints. The exact form will be dependent on the institution. However, each institution will have some unique combination of soft constraints, as policies differ from institution to institution.

4.3.1. *Soft constraints for exam-timeslot assignment*

The (*15*) common soft constraints for 'exam-timeslot assignment' (ordered by their importance):

S1) *Student-clashing*: Same as hard constraint (H2) but now it is considered as a soft constraint; formulated as using the matrices **C** and **Q**, where f_1 is a penalty function based on the number of students taking both exams e_i and e_j in the same timeslot,

$$f_1\left(\sum_{k=1}^{n_2}\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1} c_{ij}\cdot q_{ik}\cdot q_{jk}\right). \qquad (13)$$

S2) *Timeslot-capacity*: Same as hard constraint (H4) but now it is considered as a soft constraint; formulated using the matrices **E** and **Q**, where $x_c(e_i, t_j)=0$ if $\sum_{i=1}^{n_1} n_s(e_i)\cdot q_{ij}\le n_c(t_j)$, 1 otherwise, and f_2 is a penalty function based on the number of violated timeslot-capacity,

$$f_2\left(\sum_{j=1}^{n_2} x_c(e_i,t_j)\right). \qquad (14)$$

S3) *Preassigned-timeslot*: Same as hard constraint (H5) but now it is considered as a soft constraint; formulated using the matrices **A** and **Q**, where f_3 is a penalty function based on the number of assigned exams that violated the matrix **A**,

$$f_3\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} a_{ij}\cdot(1-q_{ij})\right). \qquad (15)$$

S4) *Restricted-timeslot*: Same as (H6) but now it is considered as a soft constraint; formulated using the matrices **B** and **Q**, where f_4 is based on the number of assigned exams that violated the matrix **B**.

$$f_4\left(\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} b_{ij}\cdot q_{ij}\right). \qquad (16)$$

S5) *Invigilator-exam restriction*: Same as hard constraint (H7) but now it is considered as a soft constraint; formulated using the matrices **L**, **Q** and **S**, where f_5 is a penalty function based on the number of assigned invigilators that violated the matrix **L**,

$$f_5\left(\sum_{i=1}^{n_6}\sum_{j=1}^{n_1}\sum_{k=1}^{n_2} l_{ij}\cdot q_{jk}\cdot s_{ik}\right). \quad (17)$$

S6) *Exam-proximity 1*: No student should have two exams in adjacent timeslots on the same day; formulated using the matrices **C** and **Q**,

$$f_6\left(\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1} c_{ij}\cdot prox(t_{(e_i)},t_{(e_j)})\right), \quad (18)$$

where $prox(t_{(ei)}, t_{(ej)}) = \omega$ if $|t_{(ei)}-t_{(ej)}| = 1$, 0 otherwise; $t_{(ek)}$ specifies the assigned timeslot for exam e_k, ω is a weight that reflects the penalty of violating this constraint for each student, and f_6 is a penalty function based on the total weights of students having two exams in adjacent timeslots. Most researchers used $\omega = 16$, and Cote et al. [2004] considered f_6 as the average penalty per student.

S7) *Exam-proximity 2*: No student should have two exams in s timeslots apart, $2\leq s\leq 4$, or in adjacent days; formulated using the matrices **C** and **Q**,

$$f_7\left(\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1} c_{ij}\cdot prox(t_{(e_i)},t_{(e_j)})\right), \quad (19)$$

where $prox(t_{(ei)}, t_{(ej)}) = \omega_{(i,j)}$ if $2\leq|t_{(ei)}-t_{(ej)}|\leq 4$, 0 otherwise, $\omega_{(i,j)}$ is a function that reflects the penalty of violating this constraint, and f_7 is a penalty function based on the total weights of students having two exams in s timeslots apart. Most researchers used $\omega_{(i,j)} = 2^{5-|t_{(ei)}-t_{(ej)}|}$.

S8) *Timetable-length*: The length of timetable should be minimized (equivalent to minimizing the number of free timeslots between exams); formulated using the matrix **Q**, where f_8 is a penalty function based on the total number of free timeslots between exams,

$$f_8\left(\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1}|t_{(e_i)}-t_{(e_j)}|\right). \quad (20)$$

S9) *Exam-timeslot capacity*: The total number of exams in the same timeslot should not exceed some specified number; formulated using the matrix **Q**, where $x(t_i)=1$ if $n_e(t_i)>n_{ce}(t_i)$, 0 otherwise, and f_9 is a penalty function based on the number of violated exam-timeslot capacity,

$$f_9\left(\sum_{i=1}^{n_2} x(t_i)\right). \quad (21)$$

S10) *Large exam*: Larger exams should come first to allow more time for them to be marked; formulated using the matrices **E** & **Q**, where $x_L(e_i,e_j)=1$ if $n_s(e_i)>n_s(e_j)$ & $t_{(ei)}>t_{(ej)}$, or if $n_s(e_i)<n_s(e_j)$ & $t_{(ei)}<t_{(ej)}$, 0 otherwise, and f_{10} is a penalty function that reflects the violation of this constraint,

$$f_{10}\left(\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1} x_L(e_i,e_j)\right). \quad (22)$$

S11) *Time constraints between exams*: An exam may need to be scheduled before, after or at the same time as another; formulated separately using the matrix **Q**,

$$f_{11}\left(x_T(e_i,e_j)\right), \quad (23)$$

where $x_T(e_i, e_j)=1$ if $t_{(ei)}\geq t_{(ej)}$, and 0 otherwise, if e_i needs to be scheduled before e_j ($t_{(ei)}<t_{(ej)}$),
or $x_T(e_i, e_j)=1$ if $t_{(ei)}\leq t_{(ej)}$, and 0 otherwise, if e_i needs to be scheduled after e_j ($t_{(ei)}>t_{(ej)}$),
or $x_T(e_i, e_j)=1$ if $t_{(ei)}\neq t_{(ej)}$, and 0 otherwise, if e_i and e_j need to be scheduled at the same time ($t_{(ei)}=t_{(ej)}$),
and f_{11} is a penalty function that reflects the violation of those requirements.

S12) *Same-day*: Certain exams are required to be on the same day. For instance, exams e_i and e_j need to be scheduled on the same day; formulated using the matrix **Q**, where $x_S(e_i, e_j)=0$ if $|t_{(ei)}-t_{(ej)}|\leq 1$, 1 otherwise, and f_{12} reflects the violation of this constraint,

$$f_{12}\left(x_S(e_i,e_j)\right). \quad (24)$$

S13) *Invigilator-timeslot capacity*: The total number of invigilators in the same timeslot should not exceed a specified number for each timeslot; formulated using the matrix **S**, where $x(t_i,i_k)=1$ if $\sum_{k=1}^{n_6} s_{ki} > n_{ci}(t_i)$, 0 otherwise, and f_{13} is based on the number of violated invigilator-timeslot capacity,

$$f_{13}\left(\sum_{i=1}^{n_2} x(t_i,i_k)\right). \quad (25)$$

S14) *Minimum-maximum invigilation*: The number of times of invigilation assigned to each invigilator should be within two specified numbers (minimum and maximum); formulated using the matrix **S**, where $x_M(i_k)=0$ if $n_{\min}(i_k)\le n_i(i_k)\le n_{\max}(i_k)$, 1 otherwise, and f_{14} is based on the number of violated minimum-maximum invigilation,

$$f_{14}\left(\sum_{k=1}^{n_6} x_M(i_k)\right). \quad (26)$$

S15) *Religion*: Religious convictions must be respected, e.g. Friday's prayer for Muslims. This may be satisfied by adjusting the start times of the corresponding timeslots, earlier or later, to give more time for the prayer.

4.3.2. *Soft Constraints for Exam-Room Assignment*

The (*11*) common soft constraints for 'exam-room assignment' (ordered by their importance):

S16) *Room-capacity*: Same as (H9) but now it is considered as a soft constraint; formulated using the matrices **E**, **Q** and **R**, where f_{15} is a penalty function based on the number of violated room-capacity, and $x_c(e_i, t_j, r_k) = 0$ if $\sum_{i=1}^{n_1} n_s(e_i)\cdot q_{ij}\cdot r_{ik} \le n_c(r_k)$, 1 otherwise,

$$f_{15}\left(\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_c(e_i,t_j,r_k)\right). \quad (27)$$

S17) *Preassigned-room*: Same as (H10) but now it is considered as a soft constraint; formulated using the matrices **D** and **R**, where f_{16} is based on the number assigned exams that violated the matrix **D**,

$$f_{16}\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_3} d_{ij}\cdot(1-r_{ij})\right). \quad (28)$$

S18) *Exam-room availability*: Same as (H11) but now it is considered as a soft constraint; formulated using the matrices **J** and **R**, where f_{17} is based on the number of violated exam-room availability,

$$f_{17}\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} r_{ik}\cdot(1-j_{kj})\right). \quad (29)$$

S19) *Room-utilization*: Exams are assigned to rooms in such a way that the room utilization can be maximized, or *spare seats* in each room are minimized; formulated using the matrices **E**, **Q** and **R**, where $x_U(e_i, t_j, r_k) = [n_c(r_k) - n_s(t_j,r_k)]$ if $n_c(r_k) > n_s(t_j,r_k)$, 0 otherwise, $n_s(t_j,r_k) = \sum_{i=1}^{n_1} n_s(e_i)\cdot q_{ij}\cdot r_{ik}$ represents the total number of students assigned to room r_k in timeslot t_j, $n_c(r_k) - n_s(t_j,r_k)$ represents the number of *spare seats* in room r_k in timeslot t_j, and f_{18} is based on the total number of spare seats,

$$f_{18}\left(\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_U(e_i,t_j,r_k)\right). \quad (30)$$

S20) *Exam-room capacity*: The number of exams in the same room (same timeslot) should not exceed a specified number for each room; formulated using the matrices **Q** and **R**, where f_{19} is based on the number the violated exam-room capacity, and $x_E(e_i, t_j, r_k) = 1$ if $\sum_{i=1}^{n_1} q_{ij}\cdot r_{ik} > n_{ce}(r_k)$, 0 otherwise,

$$f_{19}\left(\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_E(e_i,t_j,r_k)\right). \quad (31)$$

S21) *Similar-length*: Only exams of similar length are scheduled in the same room (in the same timeslot); formulated using the matrices **Q** and **R**, where f_{20} is a penalty function based on the total number rooms (in each of the timeslots) with at least one exam of different length, and $x_L(e_i,e_j,t_k,r_l) = 1$ if $\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1} x_{SL}(e_i,e_j)\ge 1$, 0 otherwise, $x_{SL}(e_i,e_j)=1$ if $t_{(ei)}=t_{(ej)}$, $r_{(ei)}=r_{(ej)}$, and $l_{(ei)}\ne l_{(ej)}$, 0 otherwise,

$$f_{20}\left(\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_L(e_{i_1},e_{i_2},t_j,r_k)\right). \quad (32)$$

S22) *Same-department*: Exams for the same department should be kept together, i.e. assigned to the rooms at that department; formulated using the matrices **H**, **K** and **R**, where $x_D(e_i,r_j)=1$ if $d_{(ei)}\ne d_{(rj)}$, 0 otherwise, $d_{(ei)}$ and $d_{(rj)}$ respectively indicate the departments for exam e_i and room r_j, and f_{21} is a penalty function based on the number of exams that assigned to the rooms at other departments,

$$f_{21}\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_3} r_{ij}\cdot x_D(e_i,r_j)\right). \quad (33)$$

S23) *Invigilator-room capacity*: The number of invigilators assigned to the same room (same timeslot) should not exceed a specified number for each room; formulated using the matrices **S** and **T**, where

f_{22} is a penalty function based on the number of violated invigilator-room capacity, and $x(t_i,r_j,i_k) = 1$ if $\sum_{k=1}^{n_6} s_{ki} \cdot t_{kj} > n_{ci}(r_j)$, 0 otherwise,

$$f_{22}\left(\sum_{i=1}^{n_2} \sum_{j=1}^{n_3} x(t_i, r_j, i_k)\right). \tag{34}$$

S24) *Rooms for large-exam*: When there are several groups of students taking the same exam, rooms for that exam with all student groups should be arranged near to one another; formulated using the matrices **K** and **R**, where $x_R(e_i)$=1 if $n_r(e_i)$>1, 0 otherwise, $x_D(r_j)$=0 if all $n_r(e_i)$ rooms assigned to exam e_i are located at the same department d_k, 1 otherwise, and f_{23} is based on the number of those exams, each assigned to more than one room, with at least one room located at different department,

$$f_{23}\left(\sum_{i=1}^{n_1}\left(x_R(e_i) \cdot x_D(r_j)\right)\right). \tag{35}$$

S25) *Large-room*: Large rooms should be scheduled in preference to smaller ones. The violation of this constraint may be minimized by always selecting a room with the largest capacity in the *exam-room assignment* process. The soft constraint (S10) will also contribute in minimizing the violation.

S26) *Distance*: Time should be provided for students to travel between sites. The distance traveled by students from one room in the current timeslot to another room in the next timeslot, on the same day, will be minimized if the soft constraints (S6) and (S22) are minimized.

4.4. *Examination timetabling model*

Quality measures of an examination timetable are derived from soft constraints, most frequently from student restrictions. If several different quality measures are used simultaneously, the objective function is a linear combination of these measures, with relative weights that reflect their perceived importance. If we include all of these measures, soft constraints (S1) to (S14) and (S16) to (S24), the objective function becomes very long and extremely difficult to evaluate. Similarly, if we consider all hard constraints, (H1) to (H12), a feasible timetable would hardly be produced and we would end up with *no* feasible timetable.

Since we have considered the examination timetabling problem as two separate subproblems, 'exam-timeslot assignment' and 'exam-room assignment', each subproblem will be separately modeled as a *0-1 integer programming*, as follows.

4.4.1. *'Exam-Timeslot Assignment' model*

Most of the researchers in the literature have considered only the first two hard constraints, *exam clashing* (H1) and *student clashing* (H2), and *exam proximity 1* (S6) and *timetable length* (S8) as the soft constraints, in their 'exam-timeslot assignment' models. If we consider these two hard constraints and two soft constraints, our (0-1 integer programming) 'exam-timeslot assignment' model would be

$$\text{minimize} \quad f_6\left(\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1} c_{ij} \cdot prox(t_{(e_i)}, t_{(e_j)})\right) + f_8\left(\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1} | t_{(e_i)} - t_{(e_j)} |\right) \qquad [(18) + (20)]$$

$$\text{subject to} \quad \sum_{i=1}^{n_1} x(e_i, t_j) = 0 , \qquad \text{from (1)}$$

$$\sum_{k=1}^{n_2}\sum_{j=1}^{n_1}\sum_{i=1}^{n_1} c_{ij} \cdot q_{ik} \cdot q_{jk} = 0 , \qquad \text{from (2)}$$

all variables are integers 0-1;

where $prox(t_{(ei)},t_{(ej)}) = \omega$ if $|t_{(ei)}-t_{(ej)}|=1$, 0 otherwise; $x(e_i,t_j) = 0$ if $\sum_{j=1}^{n_2} q_{ij} = 1$, 1 otherwise; c_{ij} is the number of students taking both exams e_i and e_j, $t_{(ek)}$ specifies the assigned timeslot for e_k, ω is a weight that reflects the violation penalty, f_6 is a penalty function based on the total weights of students having two exams in adjacent timeslots, and f_8 is a penalty function based on the number of free timeslots.

4.4.2. *'Exam-Room Assignment' model*

If we consider all hard constraints for 'exam-room assignment', (H9) to (H12), and the soft constraints (S20) & (S21), our (0-1 integer programming) 'exam-timeslot assignment' model would be

$$\text{minimize} \quad f_{18}\left(\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_U(e_i, t_j, r_k)\right) + f_{19}\left(\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_E(e_i, t_j, r_k)\right) \qquad [(30) + (31)]$$

$$\text{subject to} \quad \sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_c(e_i, t_j, r_k) = 0 , \qquad \text{from (9)}$$

$\sum_{i=1}^{n_1} \sum_{j=1}^{n_3} d_{ij} \cdot (1 - r_{ij}) = 0$, from (10)

$\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} r_{ik} \cdot (1 - j_{kj}) = 0$, from (11)

$\sum_{i=1}^{n_6} \sum_{j=1}^{n_2} x(i_i, t_j, r_k) = 0$; from (12)

all variables are integers 0-1;

where $x_U(e_i,t_j,r_k) = [n_c(r_k) - n_s(t_j,r_k)]$ if $n_c(r_k) > n_s(t_j,r_k)$, 0 otherwise, $n_s(t_j,r_k) = \sum_{i=1}^{n_1} n_s(e_i) \cdot q_{ij} \cdot r_{ik}$ represents the total number of students assigned to r_k in t_j, $n_c(r_k) - n_s(t_j,r_k)$ represents the number of *spare seats* in r_k in timeslot t_j, f_{18} is a function based on the total number of spare seats; $x_E(e_i,t_j,r_k) = 1$ if $\sum_{i=1}^{n_1} q_{ij} \cdot r_{ik} > n_{ce}(r_k)$, 0 otherwise, f_{19} is a function based on the total number the violated exam-room capacity; $x_c(e_i,t_j,r_k) = 0$ if $\sum_{i=1}^{n_1} n_s(e_i) \cdot q_{ij} \cdot r_{ik} \leq n_c(r_k)$, 1 otherwise, and $x(i_i, t_j, r_k) = 0$ if $\sum_{k=1}^{n_3} s_{ij} \cdot t_{ik} \leq 1$, 0 otherwise.

5. Discussion and Future Work

The GUETM, presented in Section 4, would benefit those who are involved in examination timetabling. Since different institutions have different needs, requirements and goals, this model shall be used as a guideline to develop an appropriate model for a given ETP; not some rules that every institution must follow. Many of the presented hard constraints and soft constraints will conflict either directly or indirectly. For many institutions, due to the difficulty of the problem, many are ignored altogether.

This general model has considered 12 hard constraints and 26 soft constraints. However, there remain a large number of other less importance constraints. When solving an ETP, it is advisable to start with a few number of hard constraints (2 or 3), then other hard constraints may be added one at a time until the timetable is no longer feasible. A similar approach should be used for the soft constraints. The more constraints used to solve an ETP, the more complex it would be. The formulation of the constraints as 0-1 integer programming would be much easier if more input matrices have been constructed, but more input matrices require more times and work. The *student-conflict* matrix is the most frequently used and hence the most important input matrix for ETPs.

The general model should be updated from time to time. As the number of students increased, the needs and requirements will also change and increase. Some hard constraints may need to be considered as soft constraints. As the problems changed, and new problems arise, more types of hard constraints and soft constraints should be added to the model so that it remains viable for solving a wide range of UETPs. For future work, this model will be used to develop a general model for a more complex timetabling problem, university course timetabling.

6. References

Burke, E.K., Newell, J.P., and Weare, R.F. (1995a) Examination Timetabling in British Universities: A Survey. *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling* (ICPTAT'95), 30th Aug-1st Sept, Napier University, Edinburgh, UK.

Burke, E.K., Bykov, T., and Petrovic, S. (2001) A multicriteria approach to examination timetabling. *The Practice and Theory of Automated Timetabling III, Lecture Notes in Comp. Science 2079*, pp: 118-131, Springer-Verlag.

Burke, E.K., Eckersley, A., McCollum, B., Petrovic, S., and Qu R. (2003) Using Simulated Annealing to Study Behavior of Various Exam Timetabling Data Sets. *Proceedings of the Fifth Metaheuristics International Conference* (MIC 2003), Kyoto Japan, August 25-28.

Cote, P., Wong, T., Sabourin, R. (2004) Application of a Hybrid Multi-objective Evolutionary Algorithm to the Uncapacitated Exam Proximity Problem. *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling* (PATAT2004), Pittsburgh, pp: 151-168.

Erben, W., and Song, P.Y. (2004) A Hybrid Grouping Genetic Algorithm for Examination Timetabling. *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling* (PATAT2004), Pittsburgh, pp: 487-490.

Gambhava, N., and Sanghani, G. (2003) University Examination Timetabling using Genetic Algorithm. *Proceedings of the 2nd International Conference on Applied Artificial Intelligence.*

Kendall, G., and Mohd Hussin, N. (2004) Tabu Search Hyper-heuristic Approach to the Examination Timetabling Problem at University Technology MARA. *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling* (PATAT2004), Pittsburgh, pp: 199-218.

Merlot, L., Boland, N., Hughes, B., and Stuckey, P. (2002) A hybrid algorithm for the examination timetabling problem. *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling* (PATAT 2002).

Paquete, L., and Fortseca, C. (2001) A study of examination timetabling with multiobjective evolutionary algorithm. *Proceedings of the 4th Metaheuristics International Conference* (MIC 2001), pp: 149-154.

Wong, T., Côté, P., and Gely, P. (2002) Final Exam Timetabling: A Practical Approach. *Proceedings of the 2002 IEEE Canadian Conference on Electrical and Computer Engineering.*

Wong, T. Cote, P., and Sabourin, R. (2004) A Hybrid MOEA for the Capacitated Exam Proximity Problem. *Proceedings of Evolutionary Computation 2004* (CEC2004), pp: 1495-1501.